

Amendments to the Claims:

1. (Currently Amended) A method ~~comprising: of operating a computing device comprising arranging for~~
causing a kernel portion of an operating system ~~for the computing device to~~
retrieve a property published within a first process; and
causing the kernel portion to notify the retrieved property to one or more further
processes requesting to subscribe to the property.
2. (Currently Amended) A method according to claim 1 wherein the operating
system is ~~arranged~~ configured to supply a retrieved property in the form of a first part
comprising a property name space and a second part comprising a property type.
3. (Original) A method according to claim 2 wherein the name space comprises a
category part and a key part.
4. (Original) A method according to claim 3 wherein the category part comprises a
unique identifier (UID).
5. (Currently Amended) A method according to claim 3 wherein the key part
comprises a unique identifier (UID).
6. (Currently Amended) A method according to claim 3 to wherein the name space
comprises a 64-bit integer ~~made~~ comprised of two 32-bit parts.
7. (Original) A method according to claim 2 wherein the property type comprises an
integer value and/or a byte array descriptor.
8. (Original) A method according to claim 7 wherein the byte array descriptor is of
variable length.

9. (Original) A method according to claim 7 wherein the integer value comprises 64.
10. (Original) A method according to claim 7 wherein the byte array descriptor comprises of between 0 and 512 bytes.
11. (Original) A method according to claim 7 wherein the byte array descriptor is provided in the form of Unicode text.
12. (Original) A method according to claim 2 wherein the property type is provided with an access control policy defined when the property is created.
13. (Original) A method according to claim 12 wherein the access control policy cannot be changed after the property has been created.
14. (Original) A method according to claim 12 wherein the access control policy is defined during boot of the operating system.
15. (Original) A method according to claim 12 wherein access control policy arranges the property in a reserved category which only allows a property to be defined in that category by a process having a write-system-data capability.
16. (Currently Amended) A method according to claim 1 wherein causing the kernel portion is arranged to notify comprises causing the kernel portion to notify to the one or more further processes only that the property has changed without specifying a new value for the retrieved property, thereby enabling notification of so as to enable multiple changes in the value of the property by to be notified as a single notification.
17. (Original) A method according to claim 1 wherein the kernel portion applies a limit on the number of further processes subscribing to the property.

18. (Currently Amended) A method according to claim 1 wherein the kernel portion is configured ~~arranged~~ to define ~~[[the]]~~ an order in which the property is notified to the one or more further processes, and wherein causing the kernel portion to notify comprises causing the kernel portion to notify in accordance with the defined order.

19. (Currently Amended) A method according to claim 1, further comprising causing ~~wherein the kernel portion is arranged~~ to write the retrieved property to a memory space ~~within the computing device~~ having a size which is predefined and not determined to the size of the retrieved property.

20. (Currently Amended) A method according to claim ~~[[16]]~~ 19, further comprising causing ~~wherein the kernel portion is arranged~~ to allocate further memory space for the writing of the retrieved property ~~only if in an instance in which~~ the retrieved property cannot be accommodated in the memory space of predefined size.

21. (Currently Amended) A method according to claim 1 wherein ~~the~~ causing the kernel portion to notify comprises using ~~is arranged to use~~ a kernel thread of known priority to notify the retrieved property to the one or more further processes.

22. (Original) A method according to claim 21 wherein the kernel thread of known priority comprises a supervisor type thread of the operating system kernel.

23. (Currently Amended) A method according to claim 22, further comprising using a deferred function call queued on the supervisor type thread to notify the retrieved property to the one or more further processes.

24. (Currently Amended) A method according to claim 1 wherein the property is removable ~~arranged such that it can only be removed~~ from the operating system only by ~~[[the]]~~ a process which created it.

25. (Currently Amended) A method according to claim 24 wherein removal of a property from the operating system is controlled by a security identifier (SID).
26. (Currently Amended) A method according to claim 1 wherein retrieving and/or subscribing to the property is controlled by a security identifier (SID).
27. (Original) A method according to claim 1 wherein the property is provided with a persistence attribute.
28. (Currently Amended) A method according to claim 27 wherein the kernel portion is ~~arranged~~ configured to direct the retrieved property into persistent storage.
29. (Currently Amended) A method according to claim 1 wherein the kernel portion is ~~arranged~~ configured to commit any outstanding change to the property to storage as part of operating system shutdown.
30. (Currently Amended) A method according to claim 1 wherein the property comprises a message queue facility comprising a message and message queue ~~facility for the computing device~~.
31. (Original) A method according to claim 30 wherein the message queue is provided with a handle for enabling a message queue object to be opened by a reader and/or a writer of a message in the message queue.
32. (Original) A method according to claim 30 wherein the kernel portion limits the maximum size of message that can be placed in the message queue.
33. (Original) A method according to claim 32 wherein the maximum message size is 36 bytes.

34. (Original) A method according to claim 30 wherein the size of the message queue is fixed by a first call to open the message queue.

35. (Currently Amended) A method according to claim 30 wherein messages placed in [[a]] the message queue are provided with a priority level for sequencing messages in the message queue.

36. (Original) A method according to claim 35 wherein seven priority levels are provided for messages sequenced in the message queue.

37. (Currently Amended) A method according to claim 35 wherein messages in [[a]] the message queue having the same priority level are delivered from the message queue on a first in first out basis.

38. (Currently Amended) A method according to claim 30 wherein a wait for space facility is provided for enabling, when [[a]] the message queue is full when a call is made by a party to place a message on that message queue, the ~~said~~ message to be placed on the ~~said~~ message queue as soon as space becomes available on the message queue without the need for a further call from that party.

39. (Currently Amended) A method according to claim 30 wherein a wait for data facility is provided for enabling, when no messages are present on [[a]] the message queue when a request to retrieve a message on the ~~said~~ message queue is received from a party, a message appearing on the ~~said~~ message queue to be notified to that party without the need for a further call from that party.

40-41. (Canceled).

42. (New) An apparatus comprising a processor and a memory storing computer

program code, wherein the memory and stored computer program code are configured, with the processor, to cause the apparatus to at least:

cause a kernel portion of an operating system to retrieve a property published within a first process; and

cause the kernel portion to notify the retrieved property to one or more further processes requesting to subscribe to the property.

43. (New) An apparatus according to claim 42, wherein the operating system is configured to supply a retrieved property in the form of a first part comprising a property name space and a second part comprising a property type.

44. (New) An apparatus according to claim 43, wherein the property type is provided with an access control policy defined when the property is created.

45. (New) An apparatus according to claim 44, wherein the access control policy is defined during boot of the operating system.

46. (New) An apparatus according to claim 42, wherein the memory and stored computer program code are configured, with the processor, to cause the apparatus to cause the kernel portion to notify by causing the kernel portion to notify to the one or more further processes only that the property has changed without specifying a new value for the retrieved property, thereby enabling notification of multiple changes in the value of the property by a single notification.

47. (New) An apparatus according to claim 42, wherein the memory and stored computer program code are configured, with the processor, to cause the apparatus to cause the kernel portion to notify at least in part by using a kernel thread of known priority to notify the retrieved property to the one or more further processes.

48. (New) An apparatus according to claim 42, wherein the property is removable

from the operating system only by a process which created it.

49. (New) An apparatus according to claim 42, wherein the property comprises a message queue facility comprising a message and a message queue.

50. (New) A non-transitory memory having computer-readable program instructions stored therein, the computer-readable program instructions comprising program instructions configured to cause an apparatus to perform a method comprising:

causing a kernel portion of an operating system to retrieve a property published within a first process; and

causing the kernel portion to notify the retrieved property to one or more further processes requesting to subscribe to the property.